SKP8CMINI-15,17
Tutorial 1

Software Development Process using HEW4

Renesas Technology America Inc.

Overview

The following tutorial provides an introduction to the Mini R8C Family of SKPs. It explains the basic development environment; how to develop and debug programs using HEW (High Performance Embedded Workshop) and how to work with existing example projects. Examples shown throughout this tutorial are specific to the SKP8CMINI17. If using the SKP8CMINI-15, replace any references to the SKP8CMINI-17 with SKP8CMINI-15.

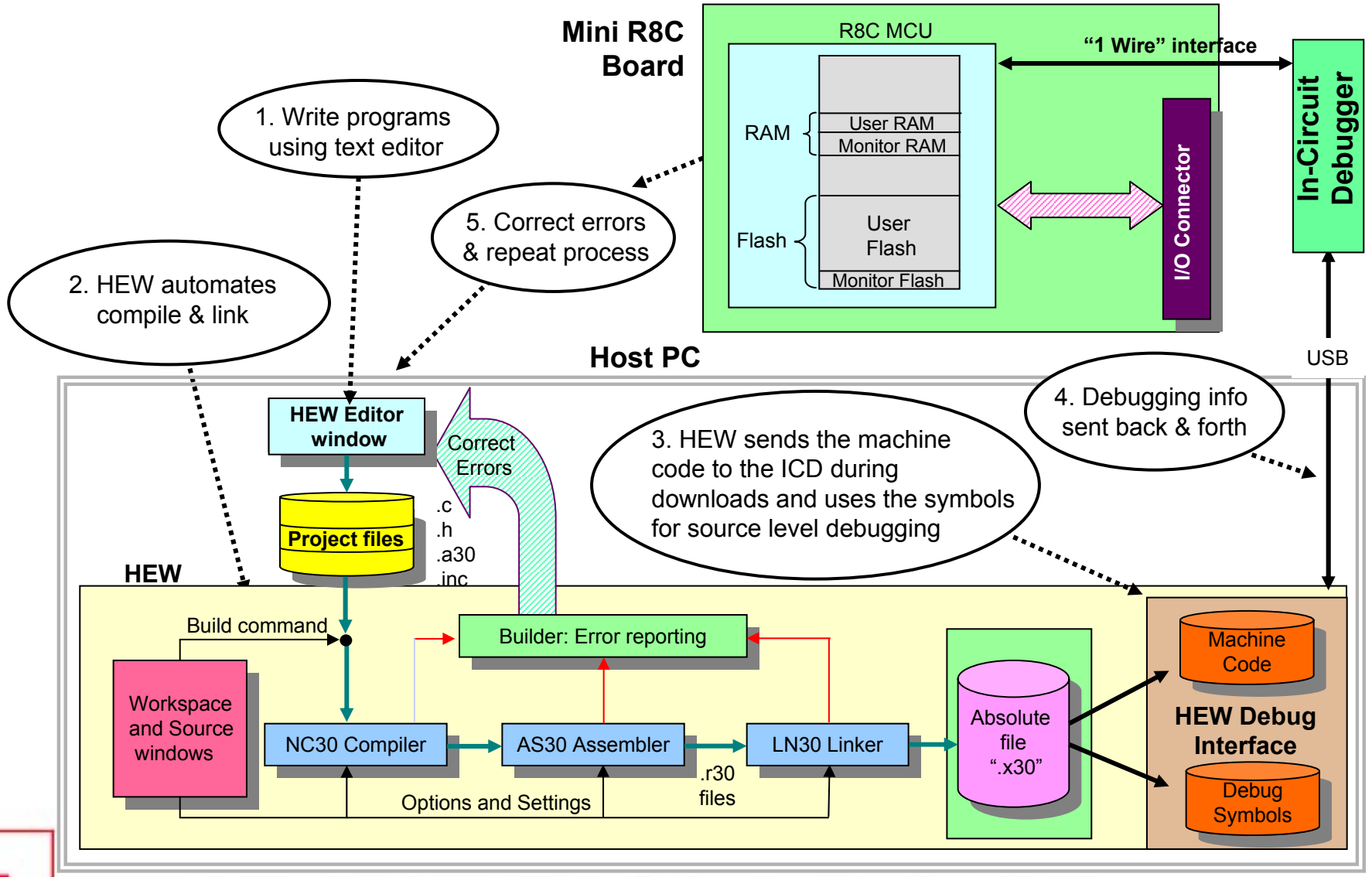
To get the most out of the Starter Kit, check out the references at the end of this tutorial.

Note: ***This tutorial assumes the user has done the following:***

1. ***Followed the 'Quick Start Guide'***
2. ***Installed the SKP files, examples, and software tools in the default directories.***

RENESAS

The Development Process

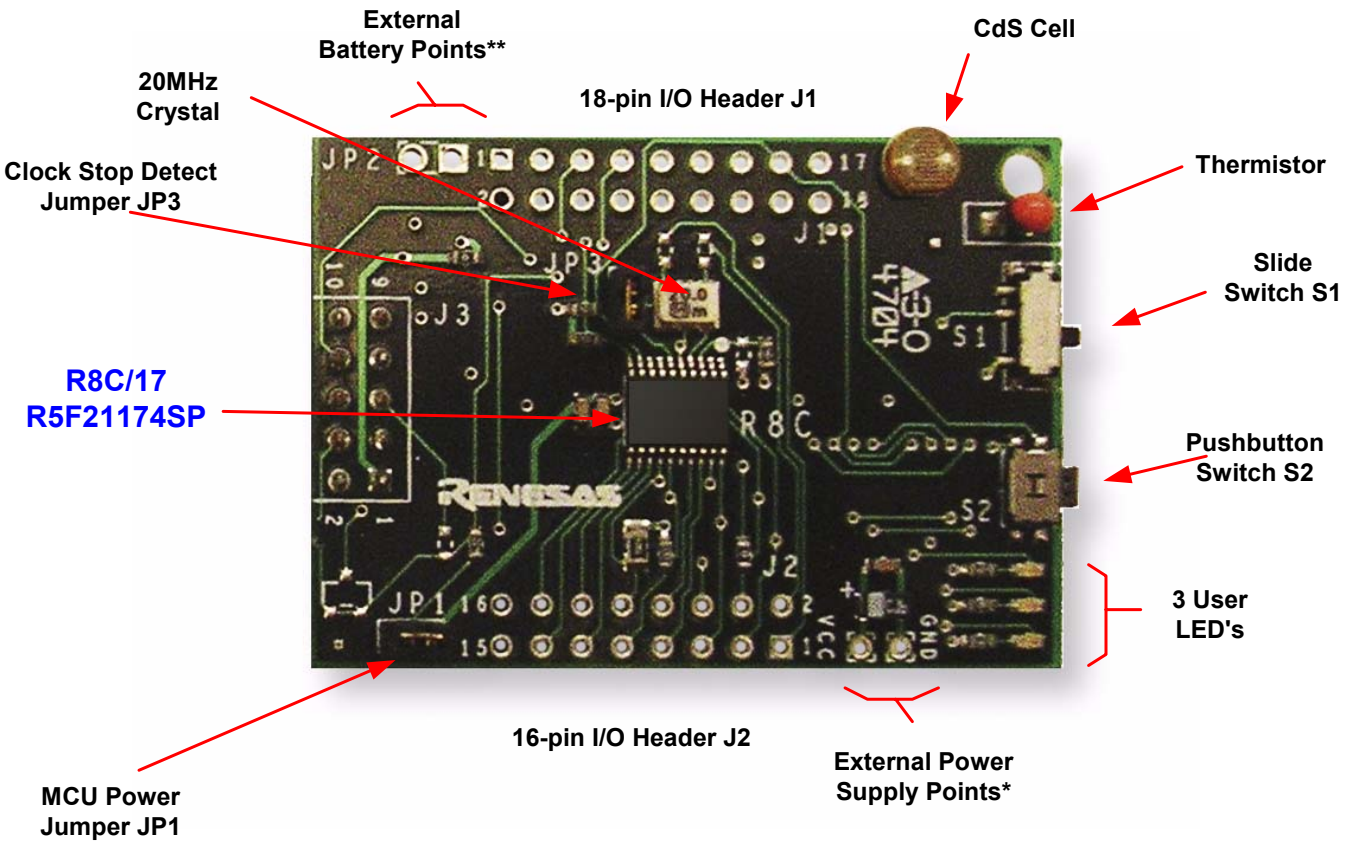


SKP8CMINI17 Connectivity

ICD (In-Circuit Debugger)

Mini R8C board

Mini R8C17 Board



R8C MCU Features

R8C/13 (R5F21134FP) MCU

- 20MHz Operating Frequency at 3.0V – 5.5V, 10MHz Operating Frequency at 2.7V – 5V
  - 16kB Flash ROM, 1kB RAM
  - 2kB x 2 Data Flash ROM
  - 24 GPIO including:
    - 4 Key-on Wakeup Inputs
    - 3 8-bit and 1 16-bit Timers plus a Watchdog Timer
    - 12-channel 10-bit ADC
    - 2 SIO – 1 Clock Sync + UART, 1 UART
    - Voltage Detect and Oscillation Stop Detection
    - Clock sources: Main (Xin), Ring oscillator (Low and High speed)

R8C/17 (R5F21174SP) MCU (same as above w/ following exceptions)

- 20 pin, 15 GPIO including:
  - 2 8-bit and 1 16-bit Timers plus a Watchdog Timer
  - 4 channel 10-bit ADC
  - 1 SIO – clock sync + UART
  - 1 I^2C channel

R8C/15 (R5F21154SP)

- same as above except I^2C channel replaced with SSU channel

See hardware manuals for product details

RENESAS

ICD (RTA-FoUSB-MON)

The ICD (In-Circuit Debugger) provides power and a USB interface to the Host PC and communicates commands and data to and from the Mini R8C board via a synchronous serial interface.

As a debugging tool (during program debug), the ICD + HEW downloads a small kernel (or ROM Monitor) program with the user program to the Mini R8C board . This kernel provides a communication interface between the R8C MCU and the ICD + HEW Debugger application on MCU status. While the kernel uses some resources of the R8C, the operation of the ICD is transparent to the user's program.

As a programming tool, the ICD + Flash-over-USB™(FoUSB) Programmer can be used to download user programs to the R8C MCU on the Mini R8C board and many other Renesas' flash MCU's (the ICD will support other Renesas flash MCU's by downloading an MCU Monitor Image (MMI) file for a particular MCU thru HEW or FoUSB Programmer).

NOTE: The kernel is only downloaded with the user program when using HEW Debugger but NOT the FoUSB Programmer.

RENESAS

Development Tools

HEW

An Integrated Development Environment (IDE) that invokes all necessary software for building and debugging your project.

NC30WA

C-compiler with Assembler. Conforms to ANSI C standards (see release notes on limitations).

Flash-over-USB™ Programmer

Flash programmer for Renesas Flash MCU's.

HEW Overview

HEW is an acronym for High-performance Embedded Workshop.

When writing a microcontroller (or any computer) program, the program is usually split into multiple files to make it easier to read and understand.

While exactly how the files are organized is up to the programmer, typically, the code is split up in a logical manner into various files (e.g. math functions in one file, serial port drivers in another, etc).

After all the files in a **project** are compiled and assembled, a **linker** combines all the files into a single file. These steps can be tedious and repetitive. To make the process simple, we use an **Integrated Development Environment (IDE**) called **HEW**.

RENESAS

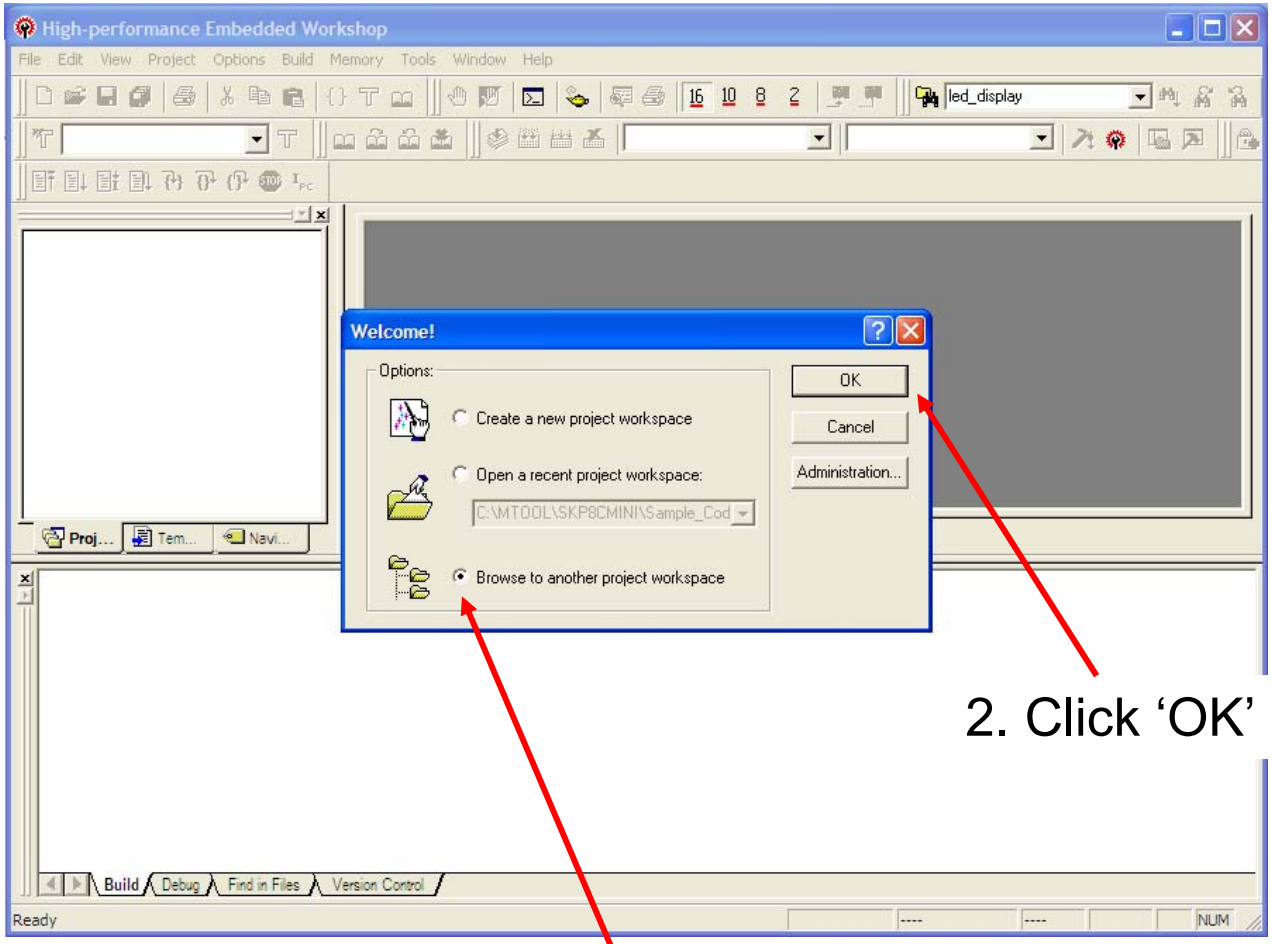
Start HEW

From the Windows Start menu, click on

Programs > Renesas > High-performance Embedded Workshop>

High-performance Embedded Workshop

Open a HEW Workspace (1/3)

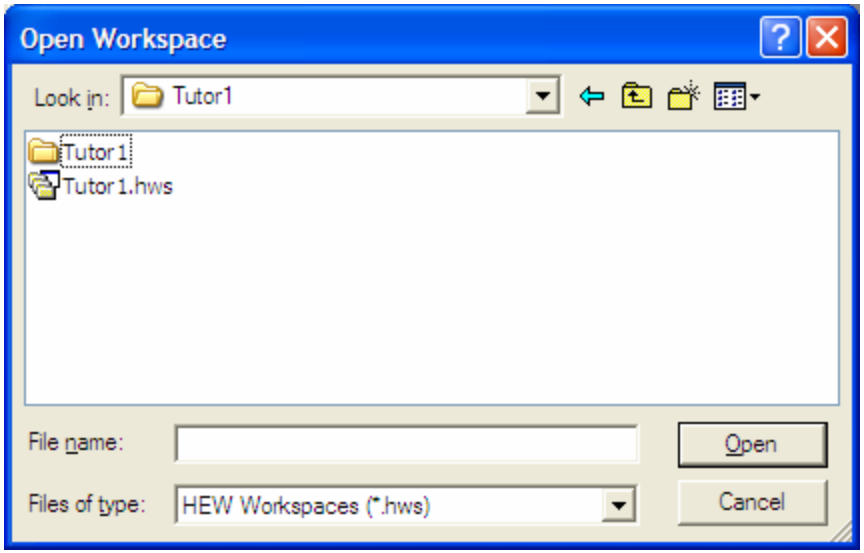


2. Click ‘OK’ button

1. After HEW opens, from the Welcome dialog box, select ‘Browse to another project workspace’ option, then click OK.

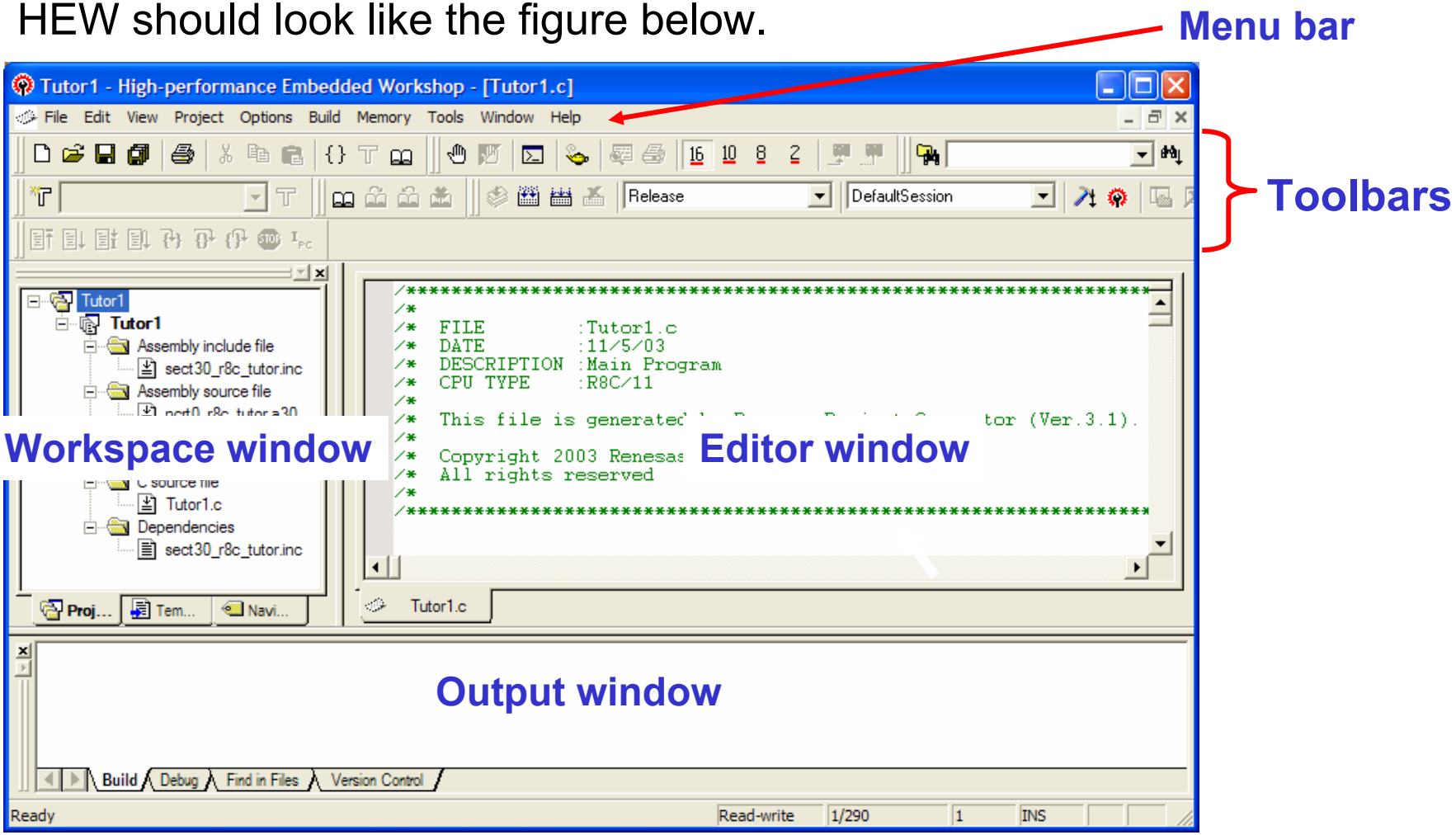
Open a HEW Workspace (2/3)

Using the Open Workspace dialog box, browse until you get to 'C:\Renesas\SKP8CMINI17\Sample_Code\Tutor1' folder. Click on Tutor1.hws HEW workspace file and then click on 'Open' button.

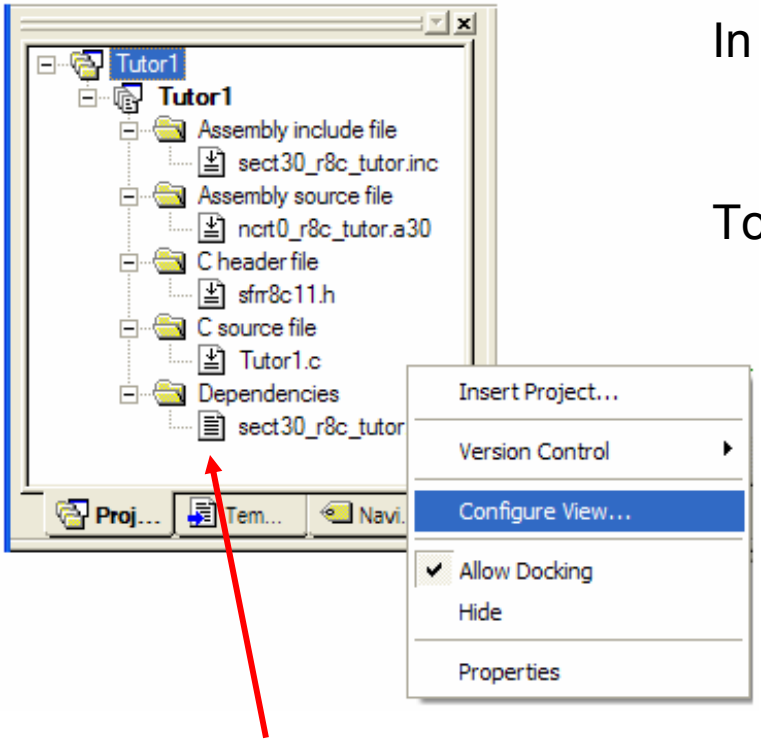


Open a HEW Workspace (3/3)

HEW should look like the figure below.



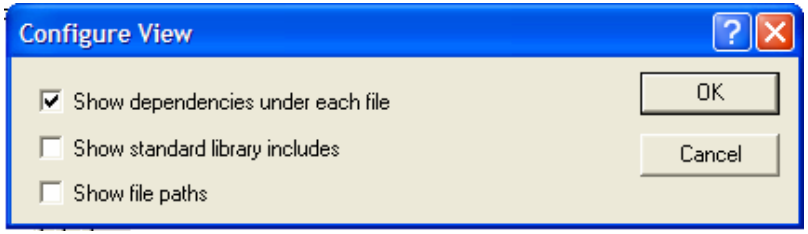
Workspace Window



To open a source file, double-click on it.

In the Projects tab, source files and header files are displayed.

To change how dependencies are displayed, e.g. show dependencies for each source file, right-click within the window, and select Configure View.



Try the following, click on 'Show dependencies under each file' and see what happens to files displayed on the window.

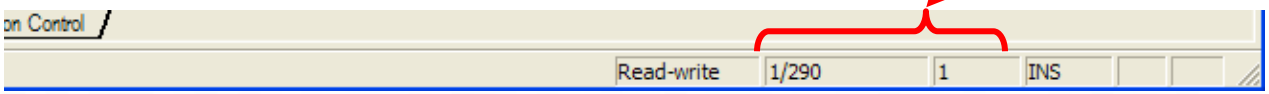
Editor (Source) Window

```
/*                                                                      */
/*  Copyright 2003 Renesas Technology America, Inc.                     */
/*  All rights reserved                                                 */
/*                                                                      */
/************************************************************************/


#include "sfrr8c11.h"   // R8C/11 special function register definitions

/* LEDs */
#define red_led     p1_0
#define yellow_led  p1_1
#define green_led   p1_2
/* Switches */
#define slider      p1_3
#define pushbutton  p4_5

#pragma INTERRUPT   tmrZ_isr
void tmrZ_isr(void);
void mcu_init(void);
void light_level_display(char);
void temperature_display(void);


char disp_count;        // LED control variable
char temp_count;        // temperature reading counter
char slider_light;      // slider sw position (light level side = 1)
int temp_value_1;       // 10-bit A/D temperature value
```

Tutor1.c

Any opened source file within the workspace are shown on the Editor window.

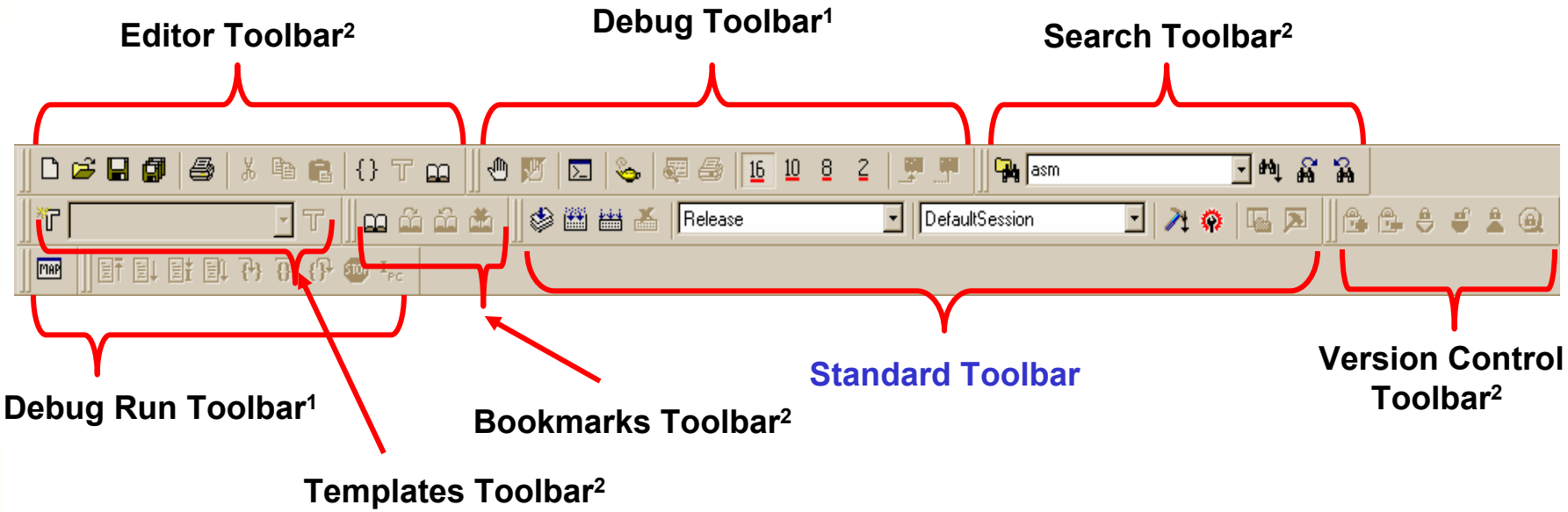
Line, total no. of lines, and column numbers are displayed here



RENESAS

HEW Toolbars

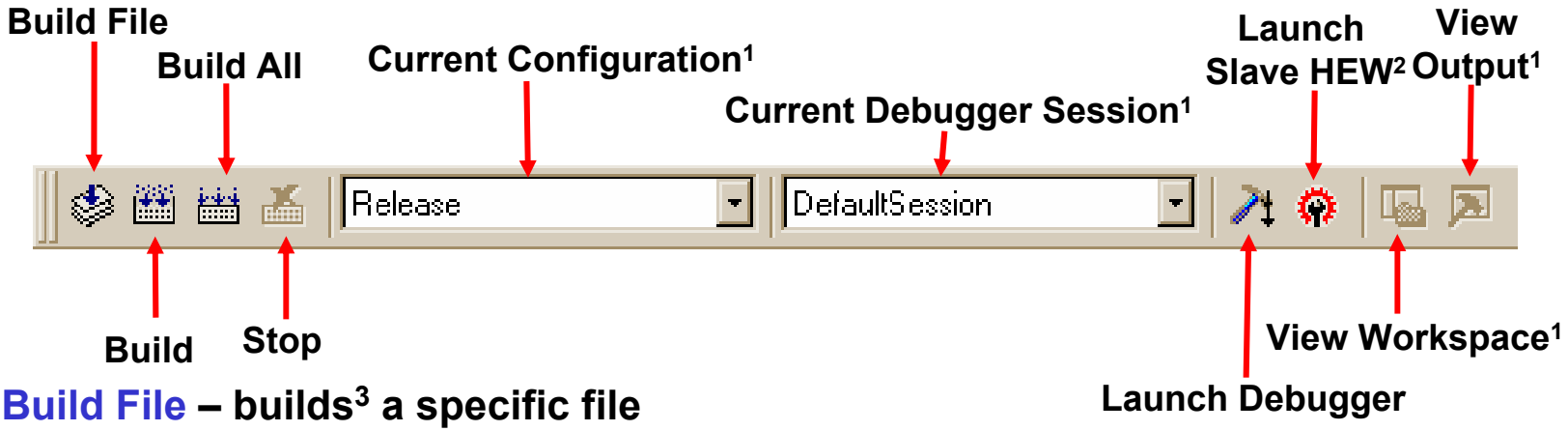
HEW is a powerful development environment with a lot of features and functionality. For this tutorial, the focus will be on features (i.e. Standard Toolbar) that will help you understand the R8C development process using HEW.



Notes:

1. ***On HEW 4.0 and above, R8C is supported by the Debug and Debug Run toolbars.***
2. ***See HEW user's manual about these toolbars.***

Standard Toolbar

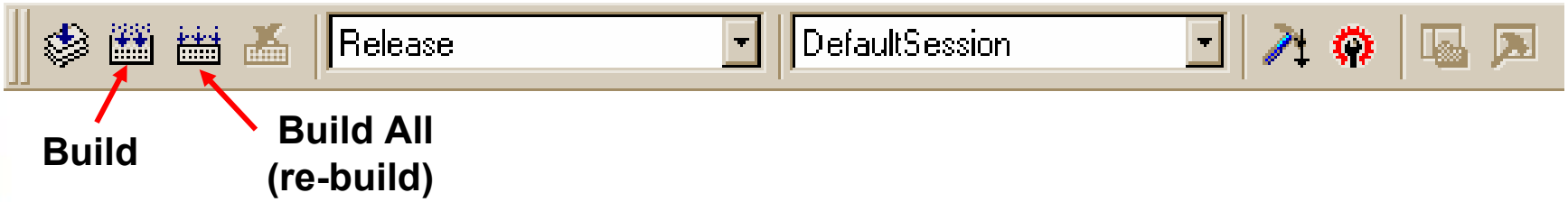


Build File – builds[3] a specific file
Build – builds files that were modified since last build
Build All – builds the whole project regardless of whether there were modifications or not
Stop – stops a running build process
Current Configuration – build configuration (e.g. for debug, optimized, etc)
Current Debugger Session – debug session configuration
Launch Debugger – calls defined debugger

Notes:

1. *These features are only supported for R8C on HEW4.0 and above.*
2. *See HEW User's manual for details.*
3. *A 'build' means running certain files (e.g. source files) under some tools (e.g. compiler, linker) to produce an output file (e.g X30 or MOT executable files for R8C)*

Build(re-build) Tutor1



Let's rebuild the Tutor1 project into an executable module, click on the 'Build All' icon. This will re-compile and link all the source files.

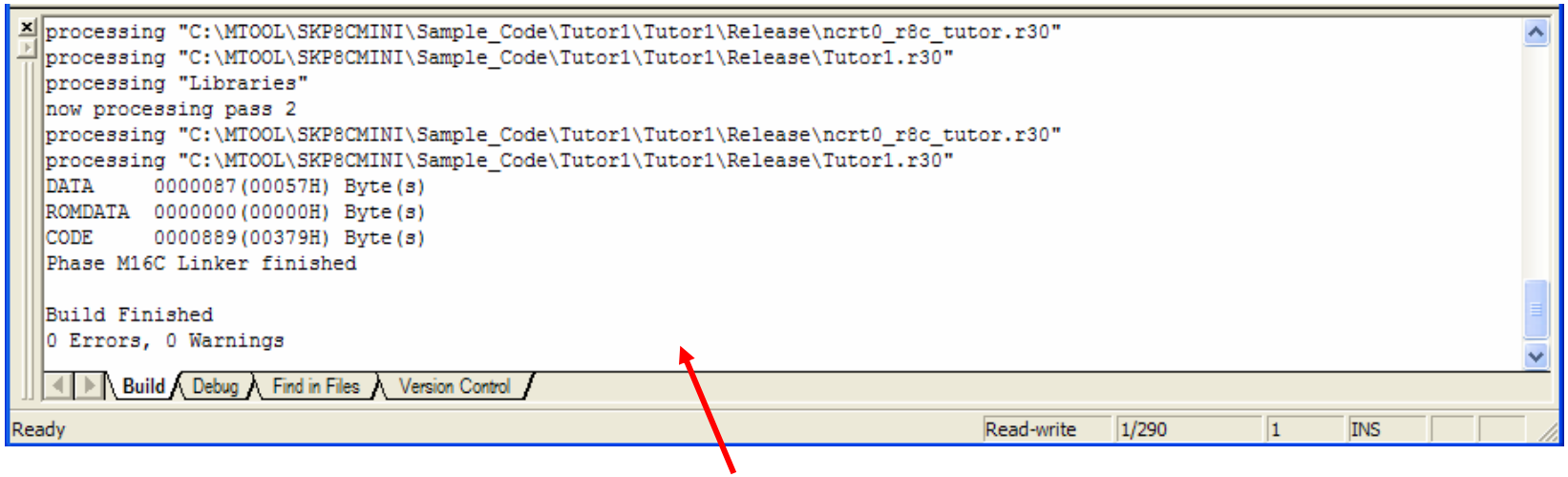
If any of the source files are modified, click on the 'Build' icon as this will only compile these modified files, which makes generating an executable module faster.

Always perform a 'Build All' when the configuration changed.

Status, errors, messages, etc during a build process is displayed on the Output window...

Output Window

The major use of the Output window is to determine if any errors or warnings occurred, and where, during the build process.



The no. of errors and warnings will show up in this window. You can then scroll up to find where the error(s) occurred. If no errors or warnings were found, 'Build Finished' will be displayed.

Now that an executable file has been created, the next step is to download and run the program on the Mini R8C board using the ICD…

HEW Debugging: Overview

The new HEW4 integrates in-circuit Debugging. These features can be used to verify that the program we developed works exactly as we intended and when it does not, we can also use HEW to find out why.

Breakpoints can be set in HEW to stop the program at certain points and verify registers, variables in memory, etc. The number of breakpoints will vary from MCU to MCU. For R8C/15 and 17 the maximum no. of breakpoints is 4.

HEW allows “step” execution in our program, which means program execution on a per line basis (whether in source level or machine code level).

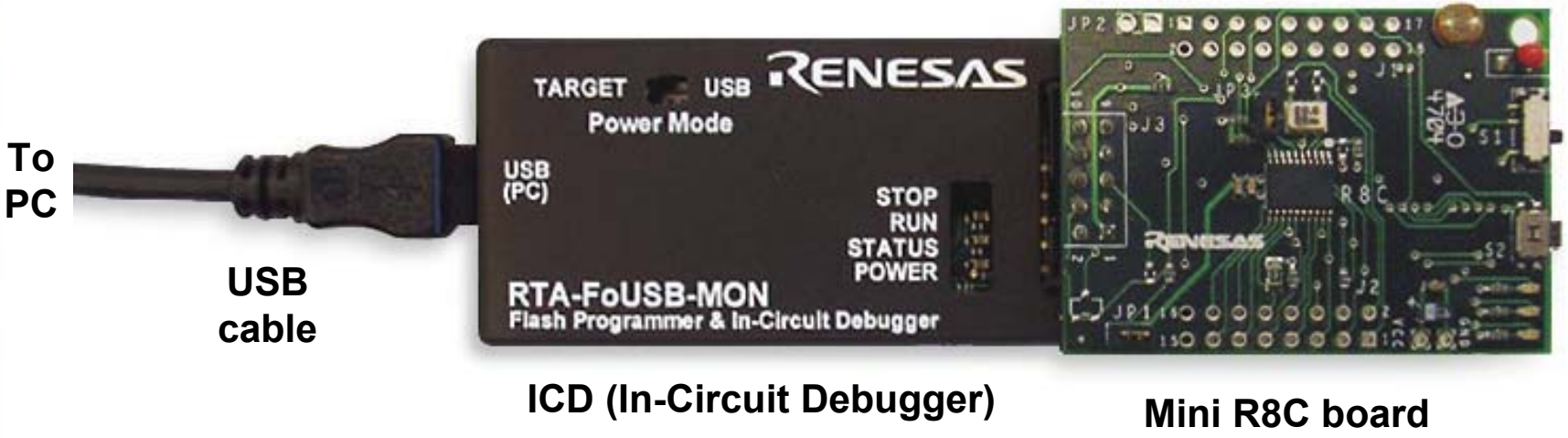
Various windows in HEW allow us to see register values and memory locations.

Debugger Exercise

- Download and run a program on the Mini R8C board
- General use of the Debugger including stepping and setting breakpoints
- Within HEW, modify the program, rebuild, and run the updated program on the Mini R8C board

Connect Hardware

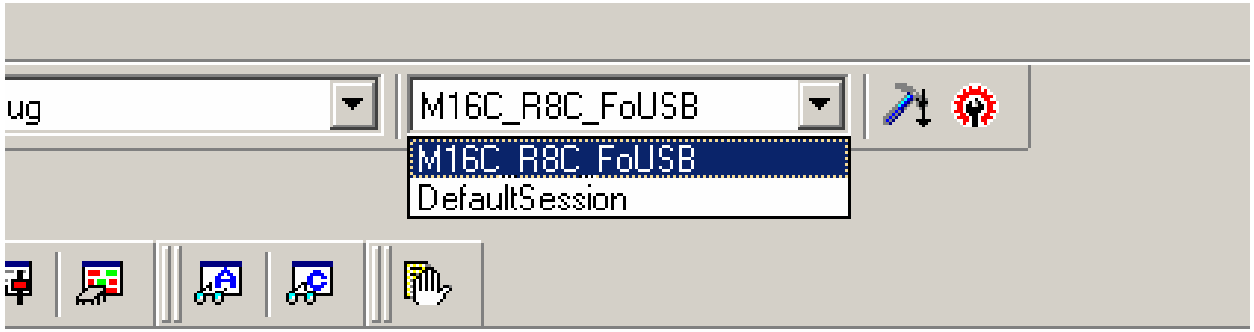
Before Debugging, connect the ICD to the Mini R8C board as shown. Connect the USB cable to the PC. On the ICD, the Power LED is on and the Status (Yellow) LED is blinking once a second (this means that the ICD USB driver was loaded correctly by Windows™).



Note: The Mini R8C board connector is not keyed, so pay close attention when connecting to the ICD.

HEW debugging example

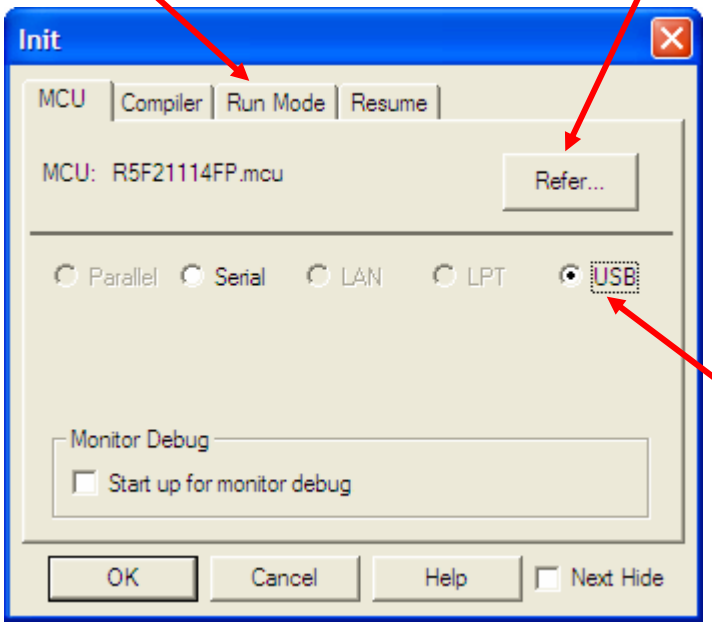
Launch a debugging session by pulling down the menu in the “sessions” box and select M16C_R8C_FoUSB.



Debug Init Window (1/2)

Step 1. Click on 'Refer..' and select 'R5F21174SP.MCU' for the SKP8CMINI-17 or 'R5F21154SP.MCU' for the SKP8CMINI15.

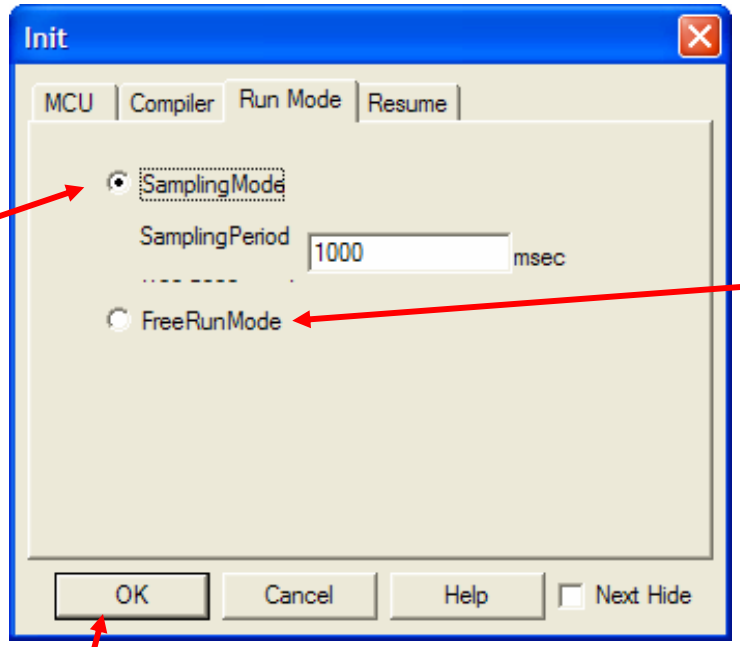
Step 3. Now click the 'Run Mode' tab



Step 2. Select USB

Debug Init Window (2/2)

For full debugging features, be sure ‘Sampling Mode[1]’ is selected.



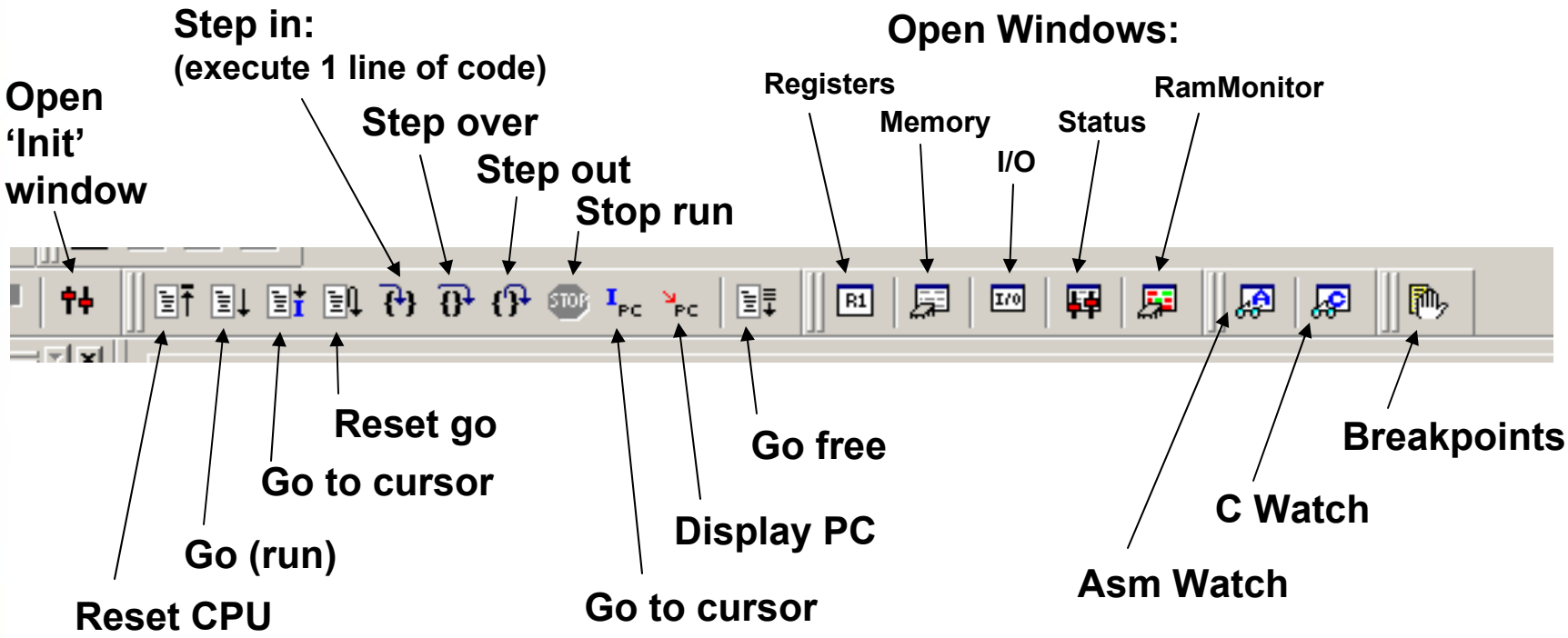
‘Free Run Mode[1]’ is for real time execution of your program, but debugging is limited. Do NOT select for this tutorial.

Now click ‘OK’ to start a HEW debug session (be sure hardware is connected). If you get an error, check all connections. See SKP user’s manual on ‘Troubleshooting’ for details.

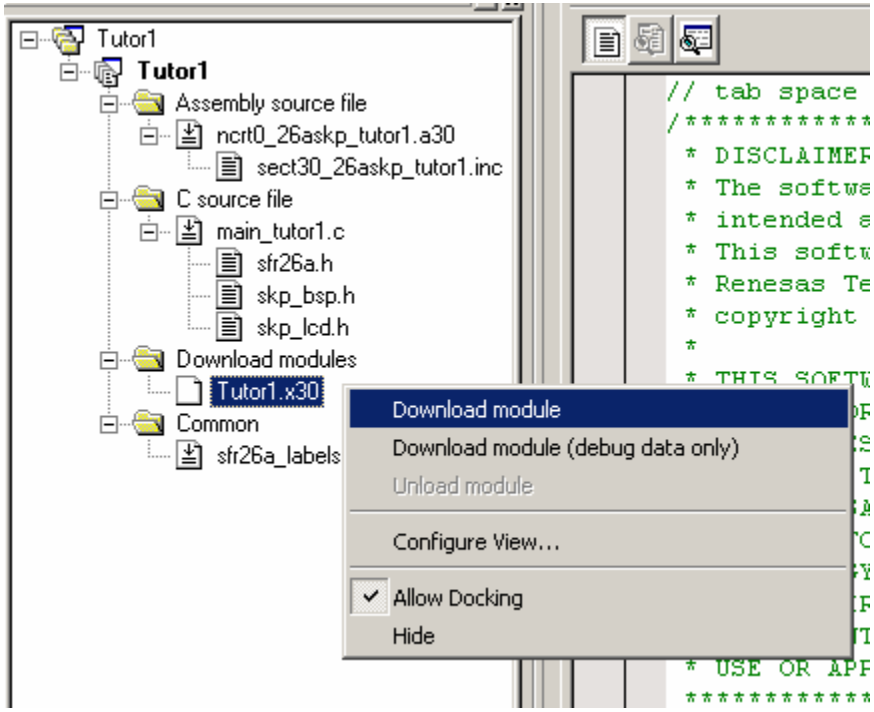
Note 1. See HEW User’s Manual or Help for the differences between Sampling Mode and Free Run Mode. Also, see the ICD (RTA-FoUSB-MON) User’s Manual for details on how ICD works under these two modes.

HEW Debug Session

Edit window has not changed, but more buttons are available in the toolbar.



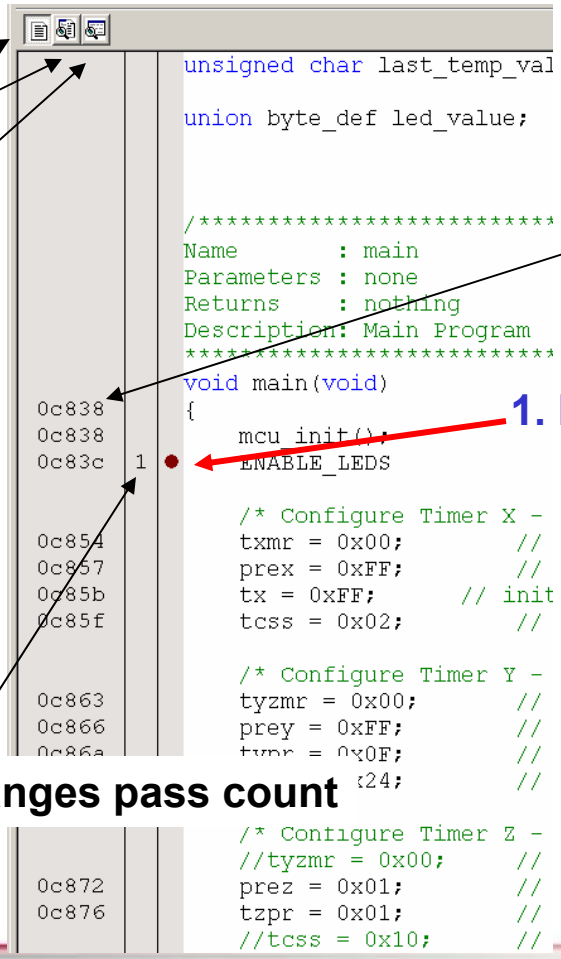
Download the executable file to the R8C/Tiny SKP Board (1/2)



In the project window, RIGHT Click on the “tutor1.x30” file, then select ‘Download Module’…

Download the executable file to the R8C/Tiny SKP Board (2/2)

After download, 3 columns are added to the edit window



View options:

- Source
- Mix
- Disassembly

Address column

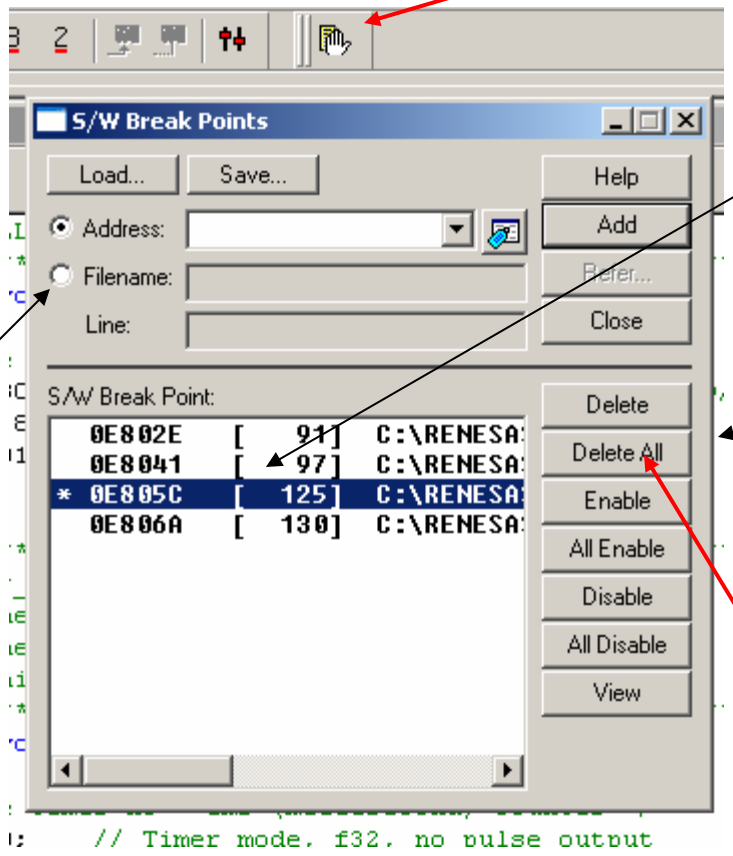
1. Double click here to set a breakpoint

Double clicking here changes pass count

2. Add a few more breakpoints….

HEW Advanced Breakpoint Options

1. Click on breakpoint button



By double clicking on an entry, you can enable or disable the Breakpoint,

or use the buttons to configure them

Breakpoint information can be saved and loaded between sessions

2. Click “Delete All”

and close window

RENESAS

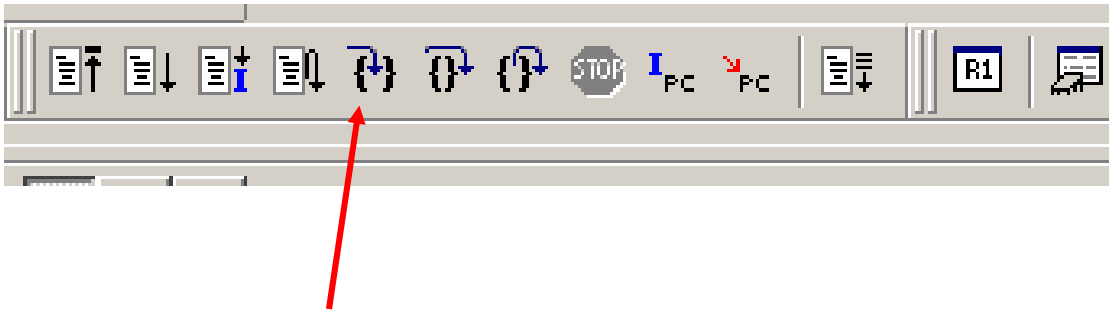
Running Downloaded Program

Click on the 'Go' icon to run the Tutor1 program you just downloaded. LED's D1, D2, & D3 will blink sequentially. Covering the CdS light cell will decrease the LED blink rate and uncovering it will increase it.

Click on the 'STOP' button.

Open the register window.

Program ‘Stepping’



Click to “step” a few lines

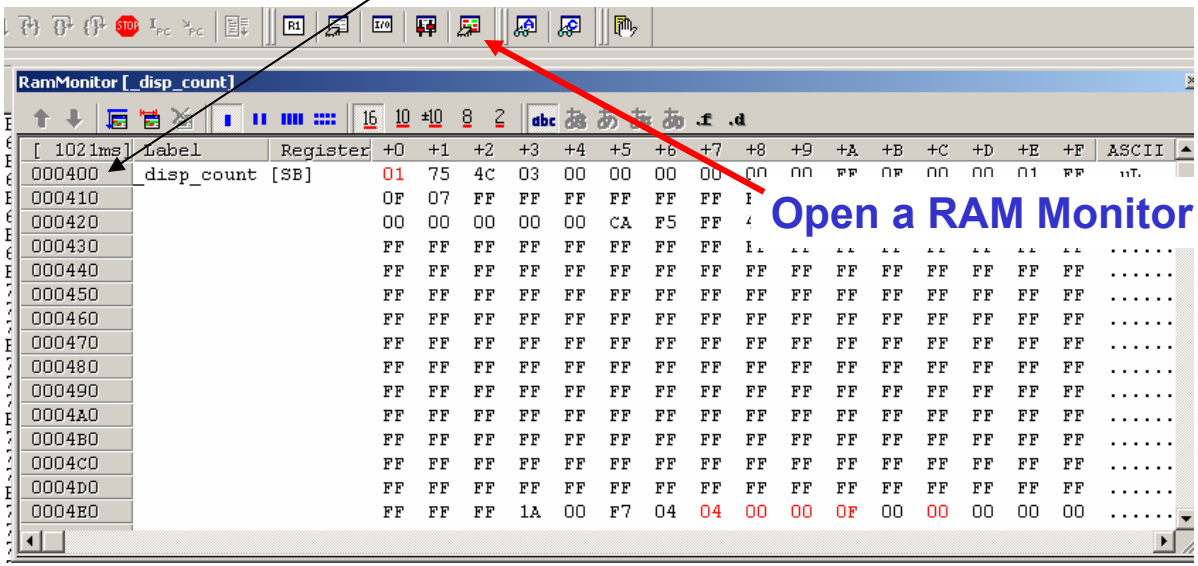
Any register values that change turn red

N...	Value	R..
R0	0001	H..
R1	0F00	H..
R2	0000	H..
R3	0000	H..
A0	0000	H..
A1	0400	H..
FB	04F7	H..
USP	0401	H..
ISP	04F7	H..
PC	0F00CE	H..
SB	0400	H..
INTB	0FF800	H..

RAM Monitor Window

The RAM Monitor displays the current value of the memory area shown on the window. It is updated at a preset value which can be modified by the user.

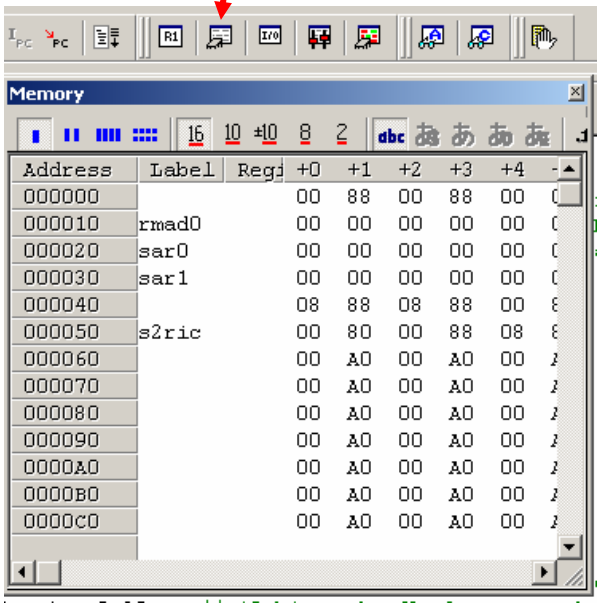
By double-click an address you can change the address range you want to view.


RamMonitor [_disp_count]
[1021ms] Label Register +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII
000400 _disp_count [SB] 01 75 4C 03 00 00 00 00
000410 0F 07 FF FF FF FF FF FF
000420 00 00 00 00 00 CA F5 FF
000430 FF FF FF FF FF FF FF FF
000440 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000450 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000460 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000470 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000480 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000490 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0004A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0004B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0004C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0004D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0004E0 FF FF FF 1A 00 F7 04 04 00 00 0F 00 00 00 00 00
Open a RAM Monitor window

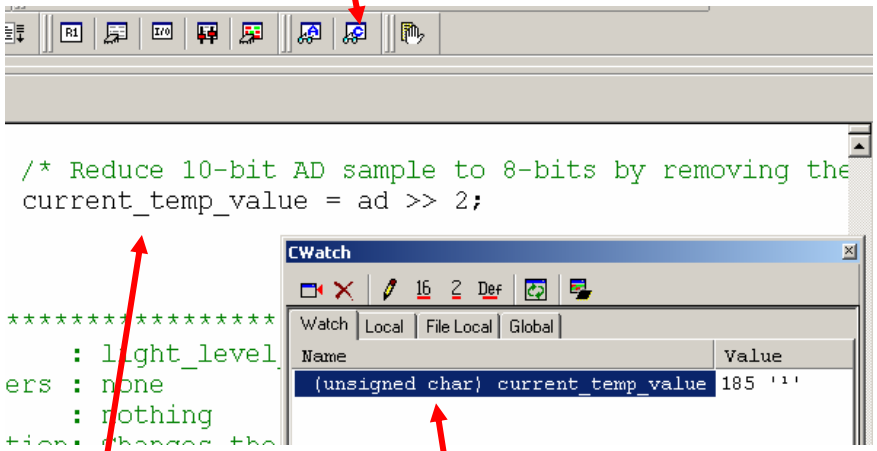
Click the 'GO' icon. You can view the RAM as it is updating. This function is not available in "Free Run" mode. Click the 'STOP' icon before proceeding. With this window open, programs do not run in real time.

Memory & C Watch Windows

Open a Memory window.



Open a C Watch window.

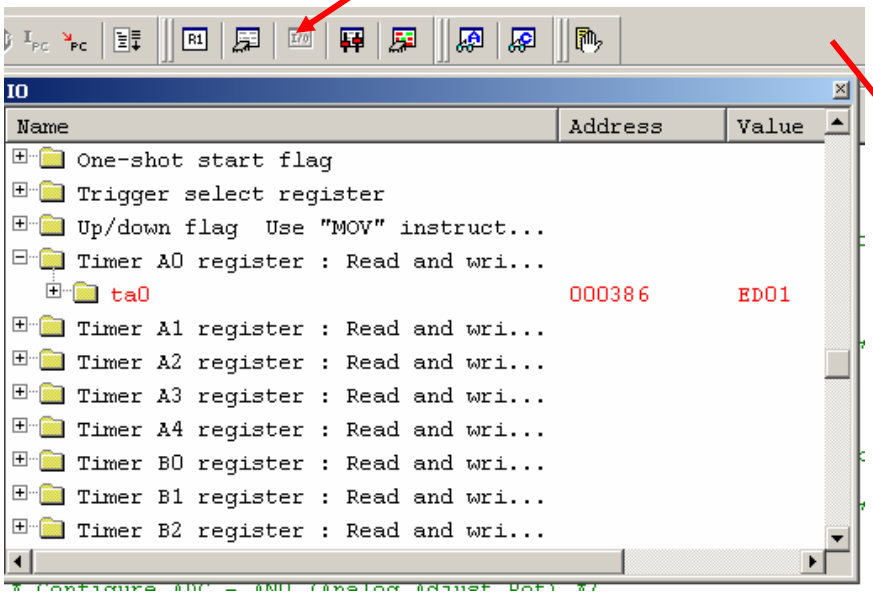

/* Reduce 10-bit AD sample to 8-bits by removing the
current_temp_value = ad >> 2;
CWatch
Watch Local File Local Global
Name Value
(unsigned char) current_temp_value 185 '¹'

The 'Memory Window' displays the location and contents of variables

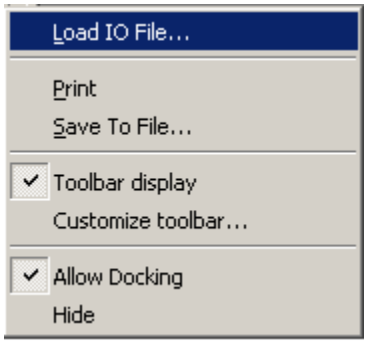
By right clicking on a variable in the file edit window, you can add it to the watch window.

I/O Window

Open the I/O window.


IO
Name
Address
Value
One-shot start flag
Trigger select register
Up/down flag Use "MOV" instruct...
Timer A0 register : Read and wri...
ta0
000386
ED01
Timer A1 register : Read and wri...
Timer A2 register : Read and wri...
Timer A3 register : Read and wri...
Timer A4 register : Read and wri...
Timer B0 register : Read and wri...
Timer B1 register : Read and wri...
Timer B2 register : Read and wri...

If the I/O window is empty, right click inside the window and "load IO File"; R8C16.io (R8C14.io for SKP8CMini15)



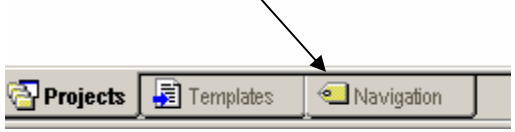
The 'I/O Window' allows you to view register contents and status of port pins

Modifying the Program (2/2)

```
void tmrZ_isr(void)
{
    /* Static Variables (retair
    static char pattern_index =

    /* Read slider switch S1 to
    if( S1 == 0 )    // Switch S
    {
        /*** MEASURE LIGHT USIN

        ch0 = 1;          // meas
        adcon1 = 0x20;    // 8-bi
        adst = 1;         // Star
        while(adst);      // wait
        tzpr = adl;       // read

        /* Advance to the next
        /* The current positior
        if( S2 )     // Button U
        {
```

1. Scroll down and find the 'tmrZ_isr' function.

(note you can also find functions using the 'Navigation" tab)



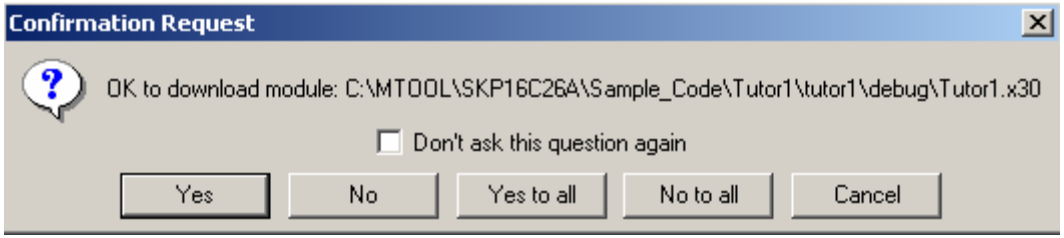
2. Change this line to 'tzpr = (0xFF – adl);'.

3. Build the project again (revised file automatically saved).

RENESAS

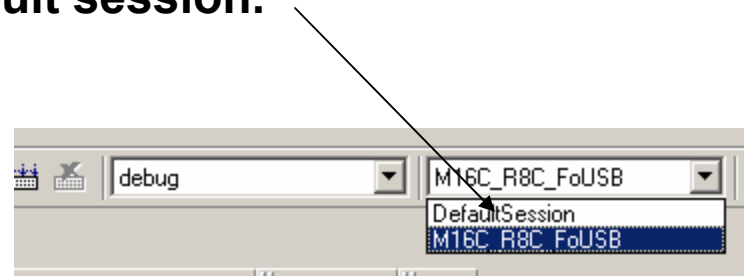
Load (re-load) Modified Program

The HEW debugger will automatically detect that the program has been recompiled and ask if you wish to download again.



Click 'Go'. Now, decreasing the light level increases the blink rate.

To stop a debug session but leave HEW open (i.e. edit only), re-open the default session.



End of Tutorial

This is the end of the tutorial. You can try downloading other sample programs from the \Sample_Code directory.

Tutorial 2 provides step by step instructions on how to use the Project Generator to simplify project creation. It also provides specific details on setting up your environment and creating a new project from scratch.

In addition, check out the references on the next page.

Have Fun!!

RENESAS

References and Recommended Reading

All documents that came with the SKP can be found using the "Document Description" from the Start > Programs > Renesas > SKP8CMINI17 (15) menu.

- **SKP8CMINI User's Manual:** This is a "must read" document! It details all the things you need to know on how to use the Starter Kit.
- **R8C Hardware Manuals:** Device specifications for R8C/Tiny MCUs.
- **HEW User's Manual:** To fully understand and get the most out of HEW, this is recommended reading.
- **NC30WA Version X.XX User's Manual:** Check this manual out for features specific to the NC30 compiler.
- **RTA-FoUSB-MON User's Manual:** Read this manual to understand how the In-Circuit Debugger / Flash Programmer works.

More References and Recommended Reading

- **M16C Series C Language Programming Manual:** This is a great document for any level of programmer. The first chapter is an introduction and reference on the C language. The next chapter explains specifics of C programming with the M16C family of microcontrollers.
- **R8C/Tiny Series Software Manual:** This document describes the instruction set and timing information for the R8C/Tiny series MCUs.
- **AS30 Version X.XX User's Manual:** Read this manual if you plan on writing programs in Assembly or when making changes to the startup file.
- **Application Notes and Sample Programs:** Application notes and other sample programs can be accessed from Renesas Technology America's website: http://www.renesas.com.
- **SKP updates:** www.renesas.com/skp.

RENESAS